

Modbus on serial line

- Being able to approach the implementation of Modbus products on Serial Line

A Operating Principles

B Modbus RTU and Modbus ASCII

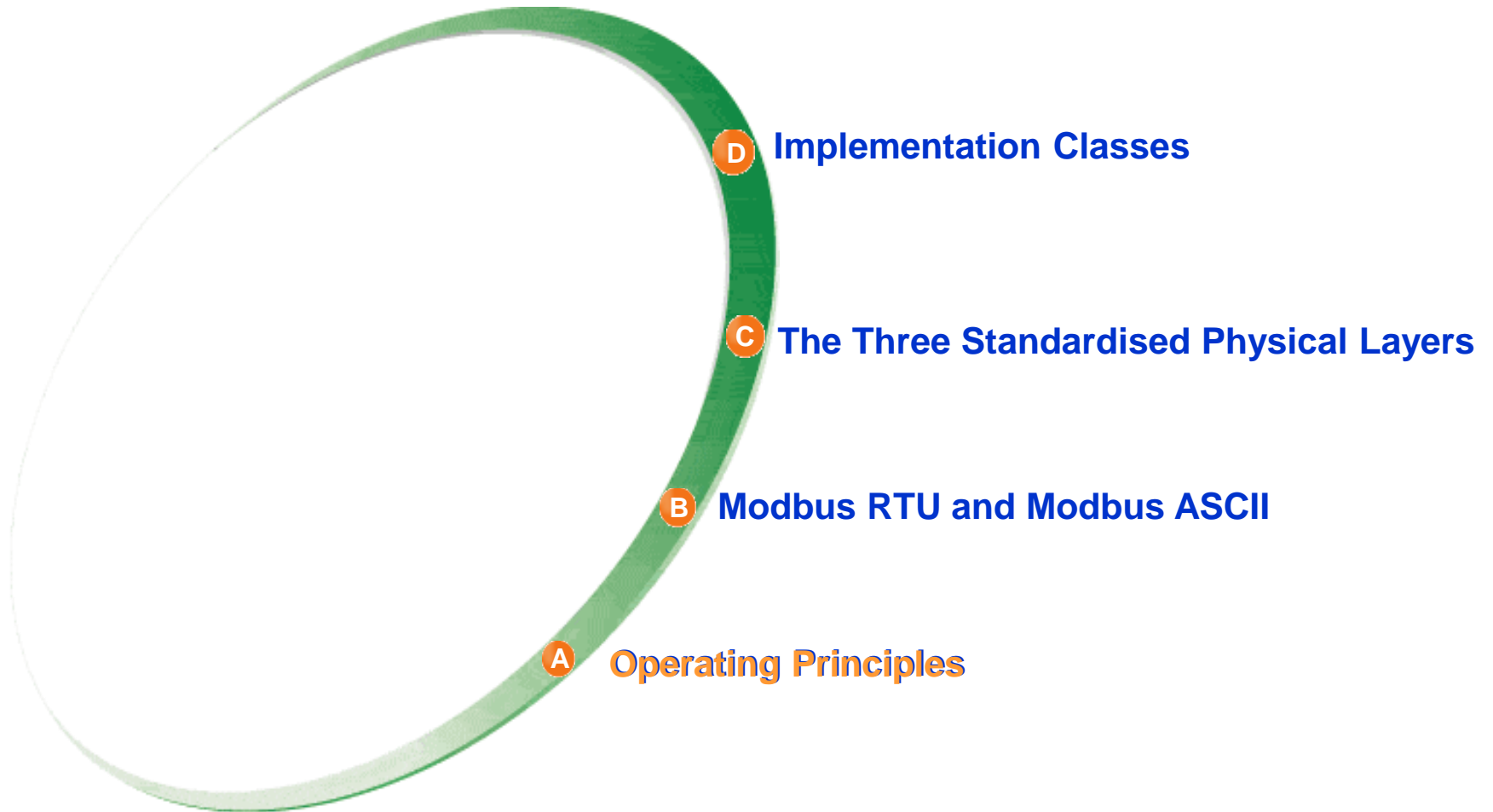
C The Three Standardised Physical Layers

D Implementation Classes

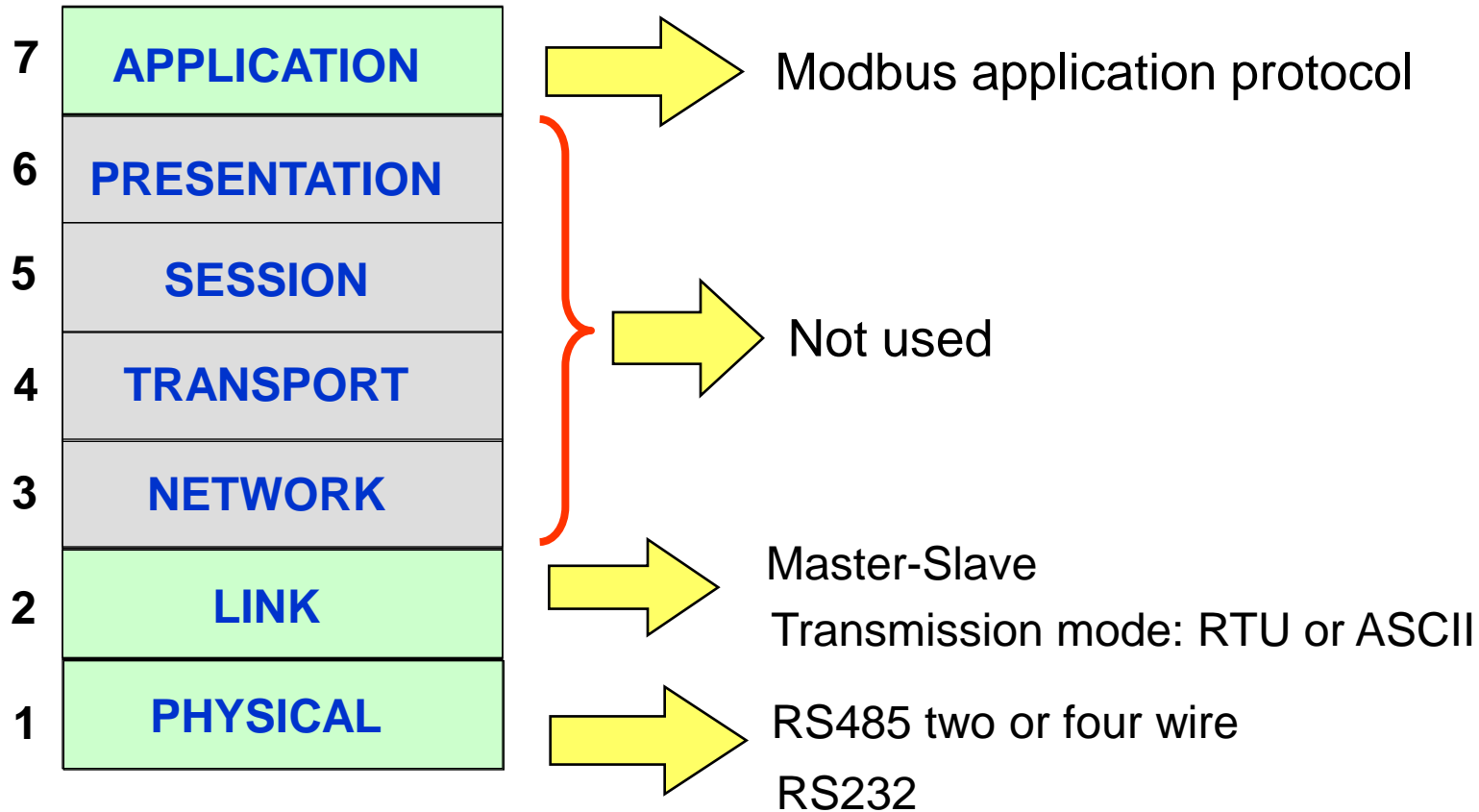
Duration : 40 min.

Training expert: Philippe WARIN
Production : Schneider-Electric

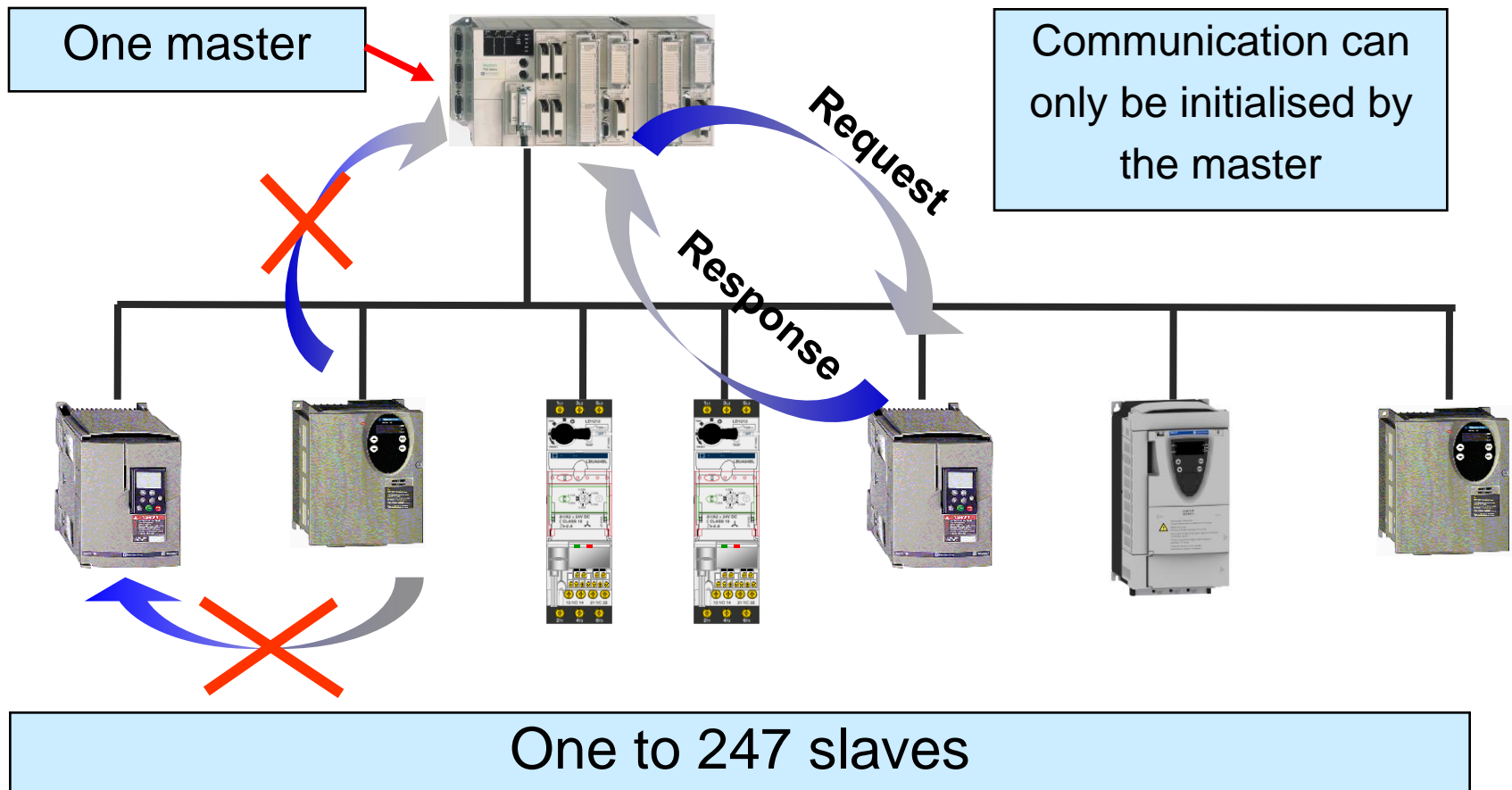
OPERATING PRINCIPLES



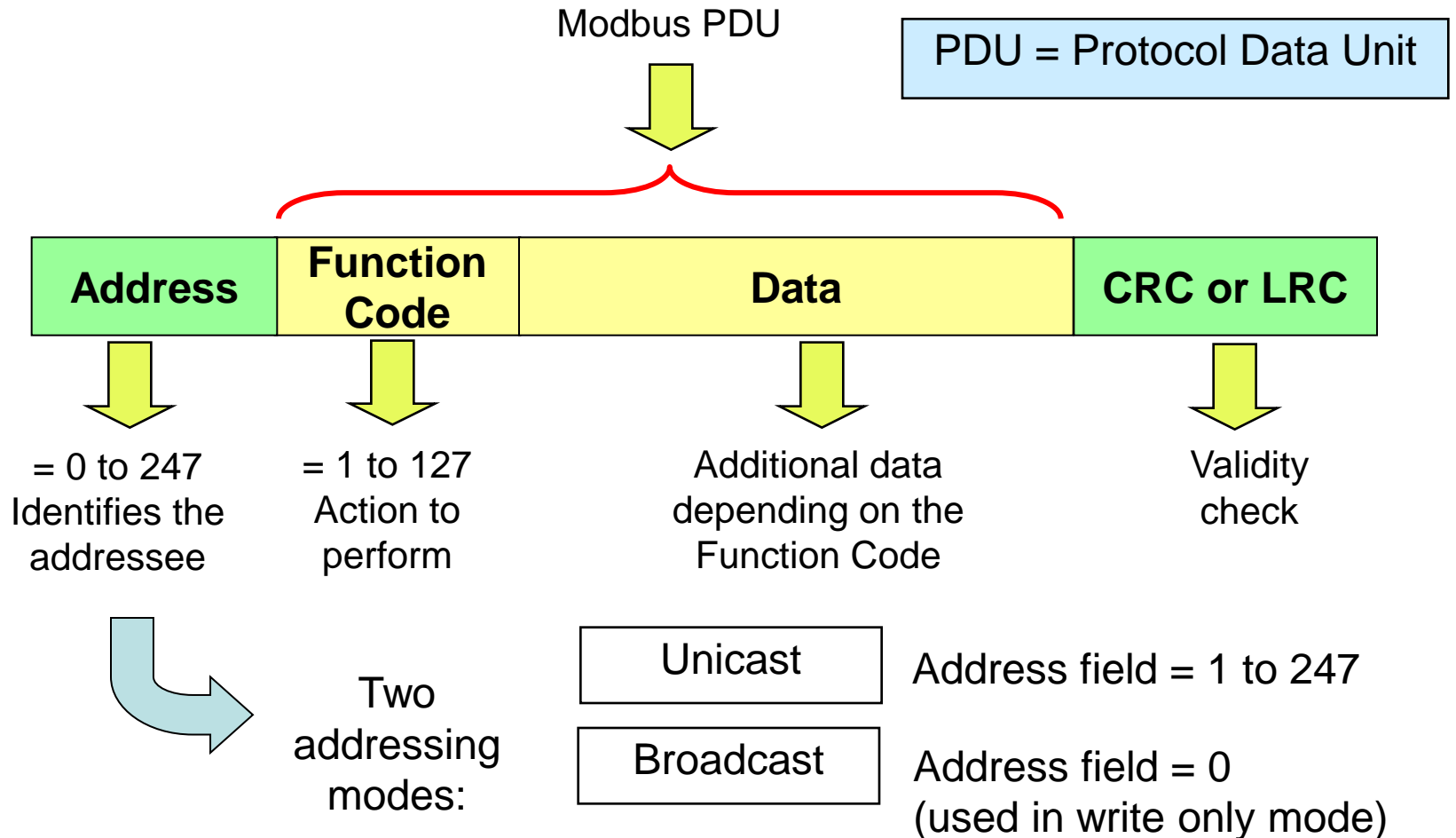
- Modbus serial line and the OSI model



- Modbus Master-Slave protocol principles

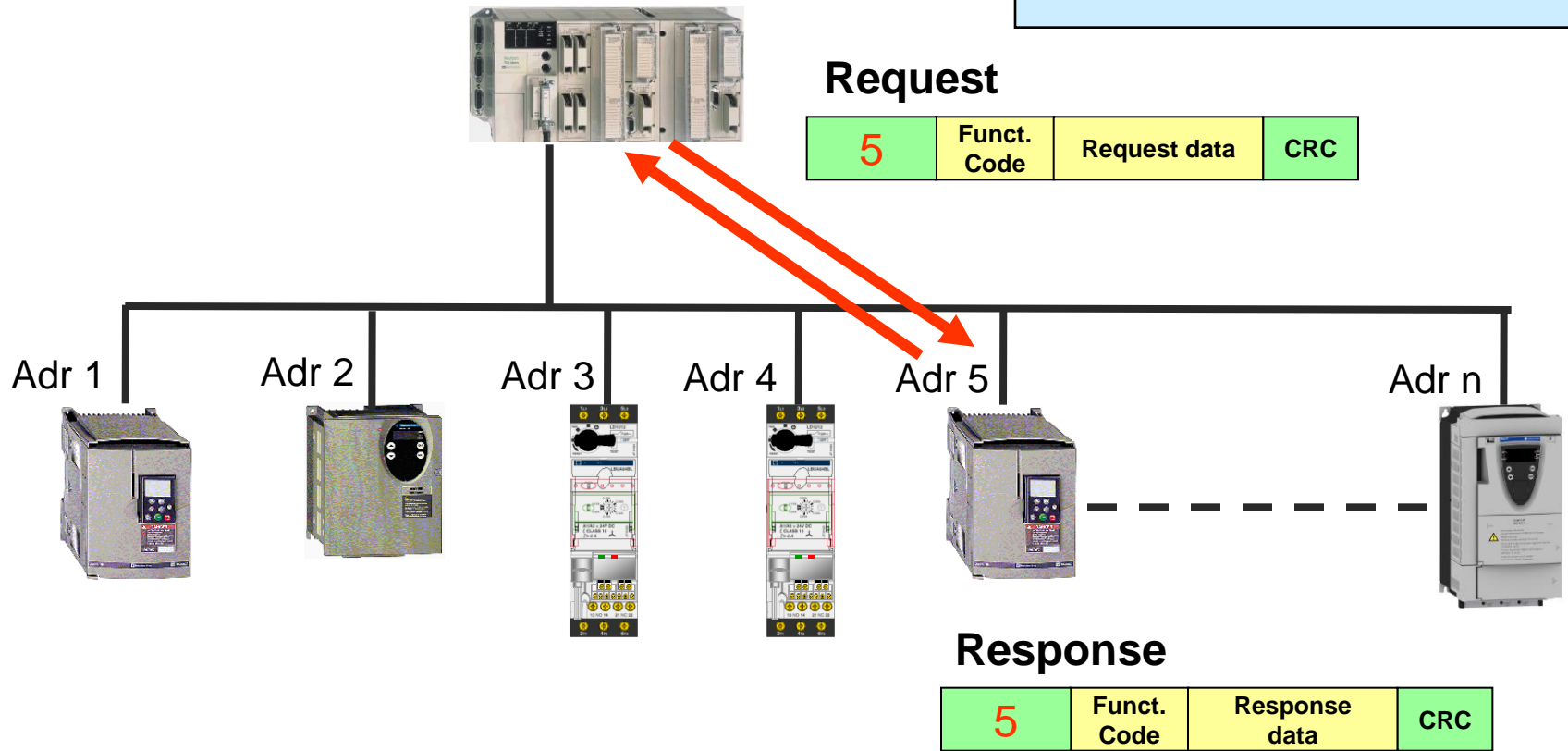


- Modbus serial line frame



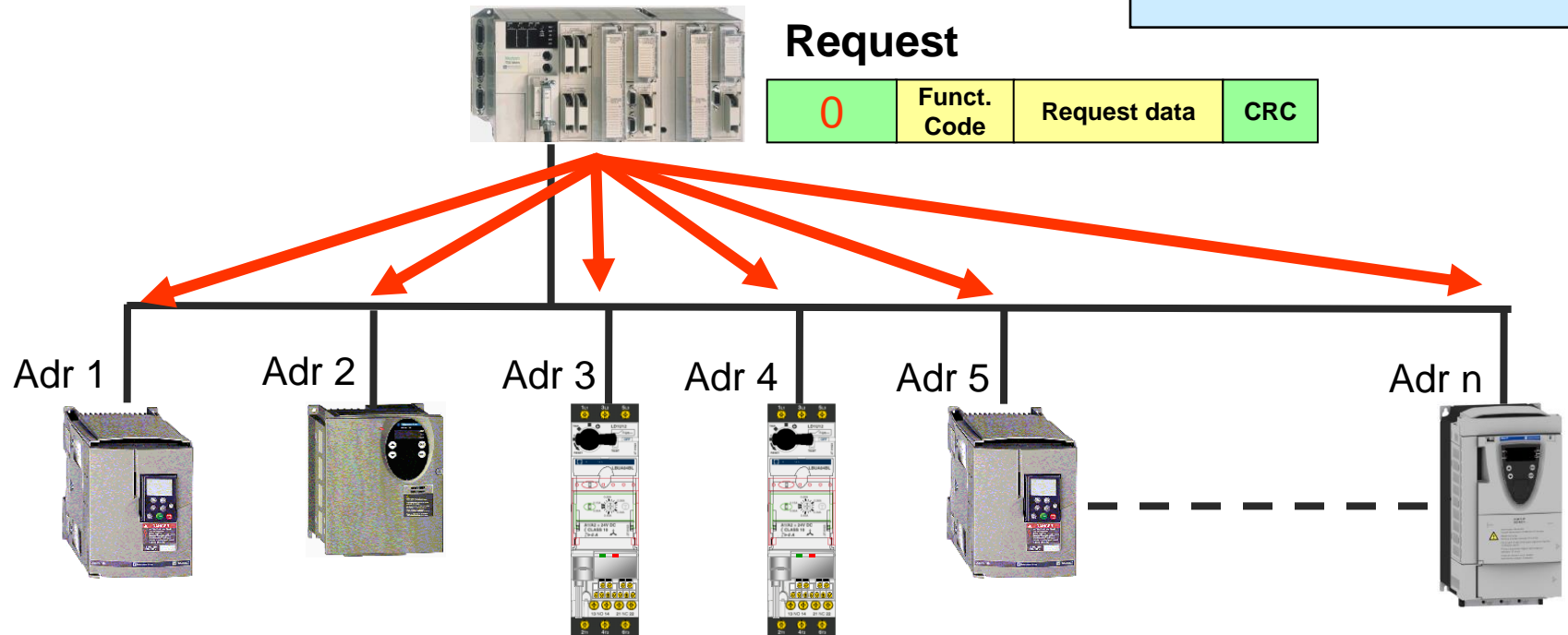
- Unicast mode addressing

Address = 1 to 247



- Broadcast mode addressing

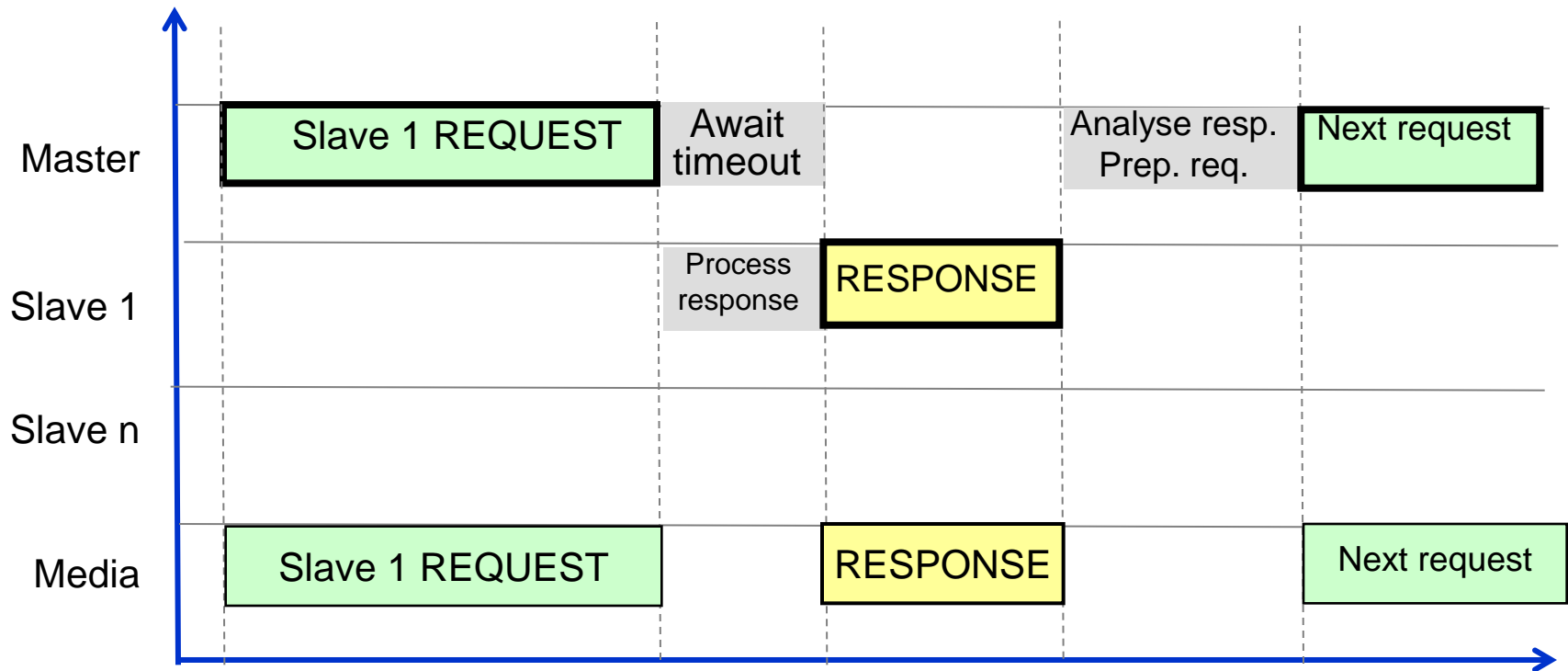
Address = 0



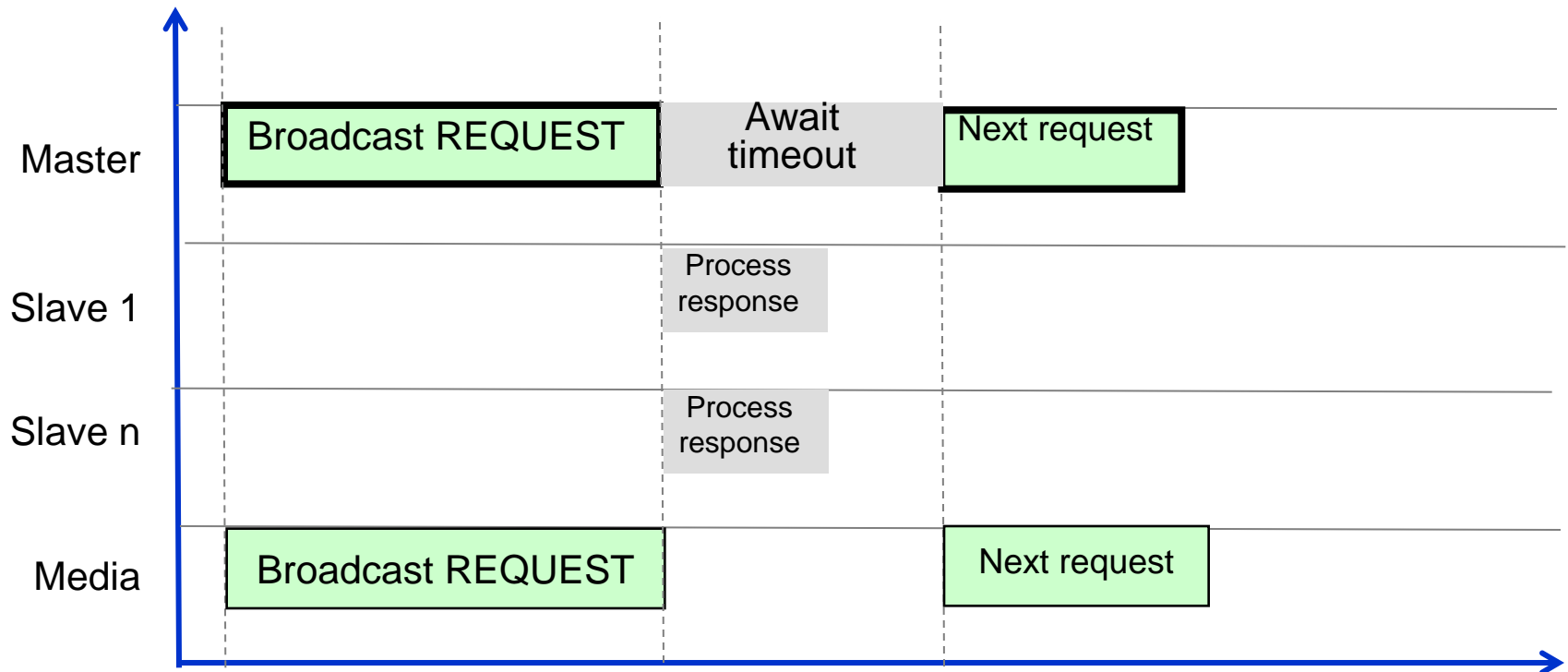
No response from slaves

Used only for write functions

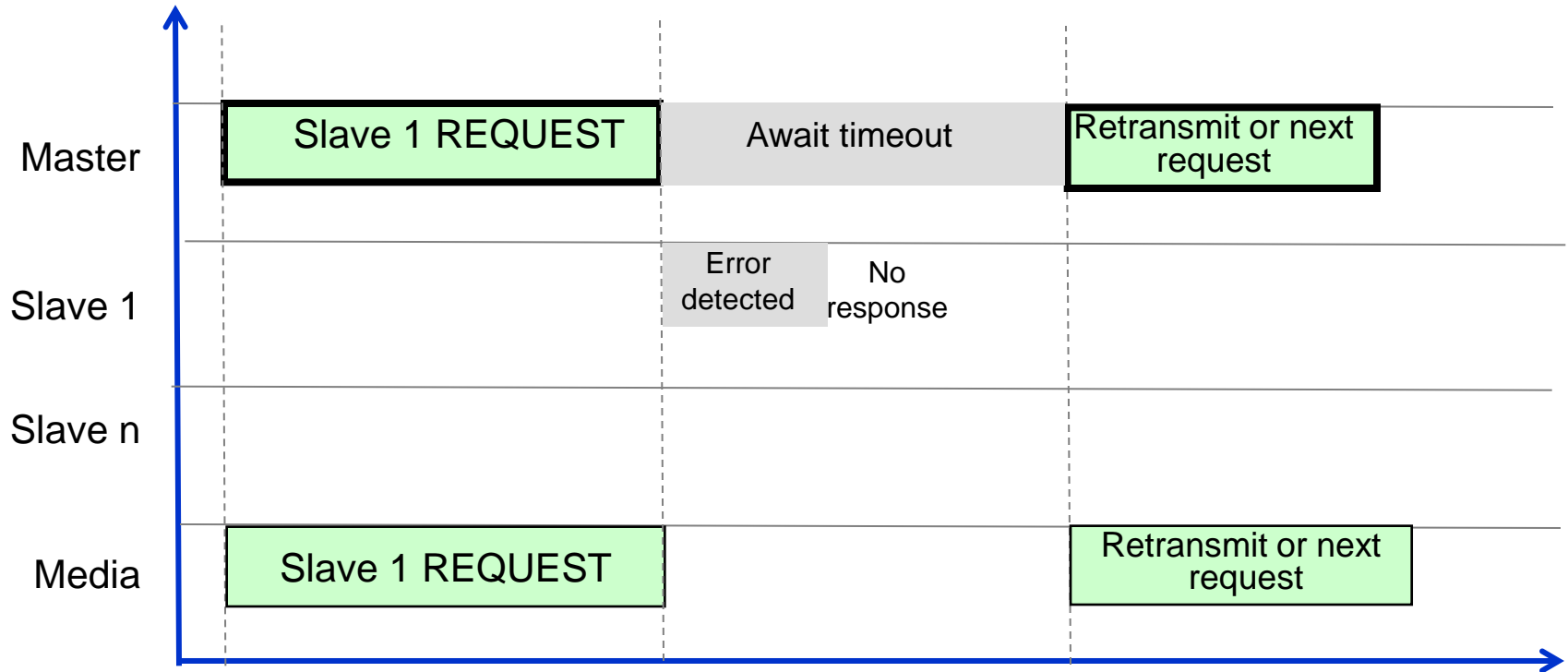
- Unicast addressing timing diagram



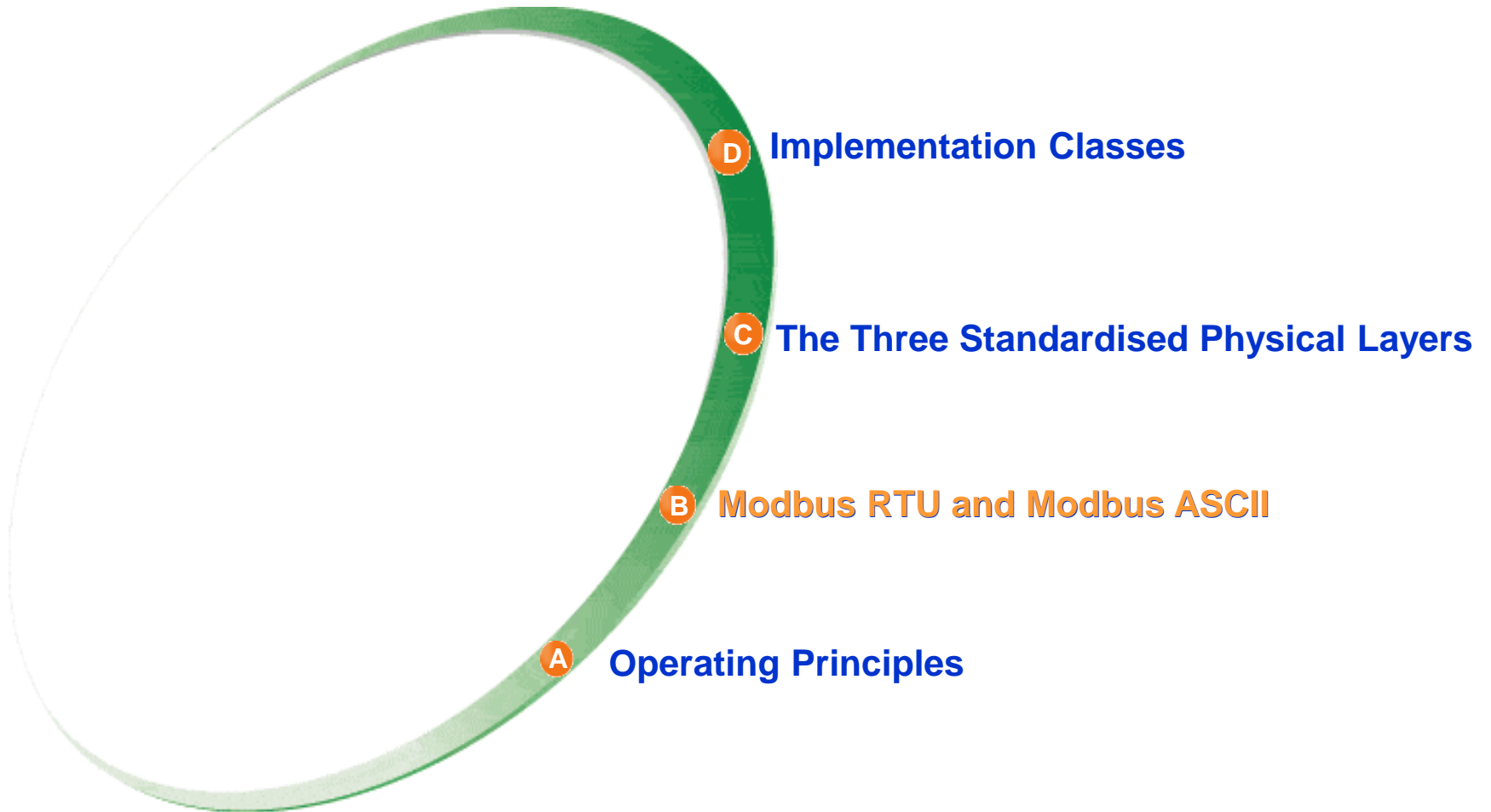
- Broadcast addressing timing diagram



- Transmission error event timing diagram



MODBUS RTU AND MODBUS ASCII



- Two defined serial transmission modes

RTU mode

Required

ASCII mode

Optional


Defines:

 the **content and format** of each character

 its **encapsulation**: the start and end of the frame

Transmission mode and serial port parameters are identical on all products.

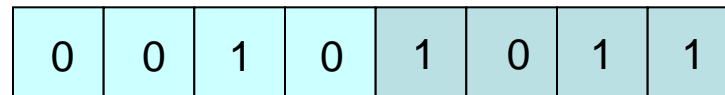
- RTU mode: content

 Every data item is encoded in **one byte**



Two hexadecimal digits

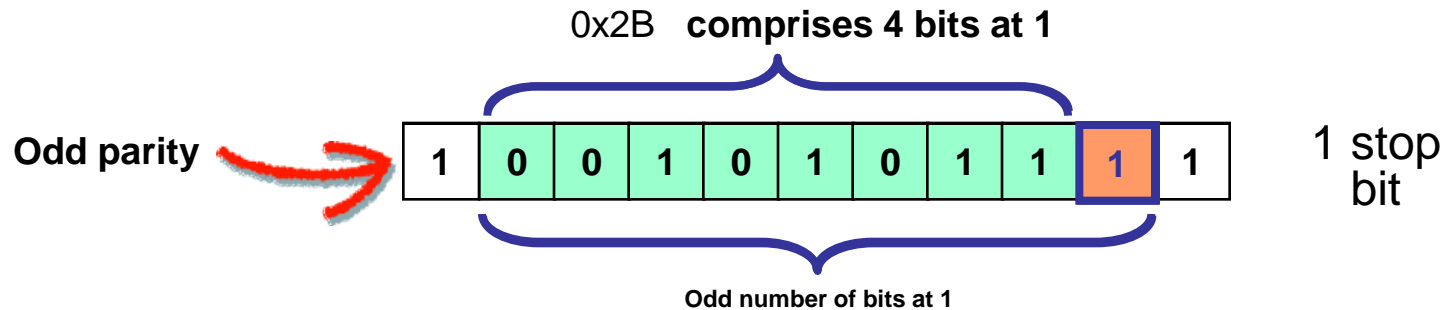
Encoding example: **Read Device Identification = 43 = 0x2B**



- RTU mode: character format = 11 bits

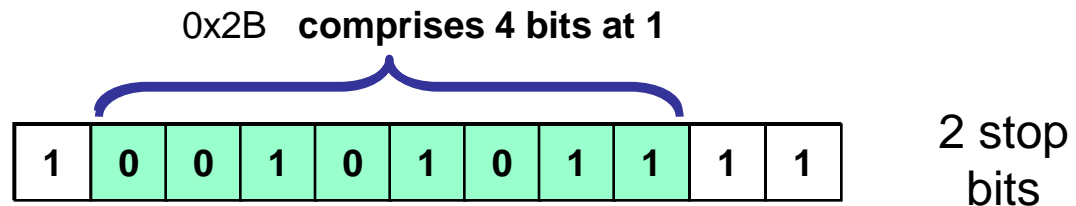
With parity check

Encoding example: **Read Device Identification = 43 = 0x2B**

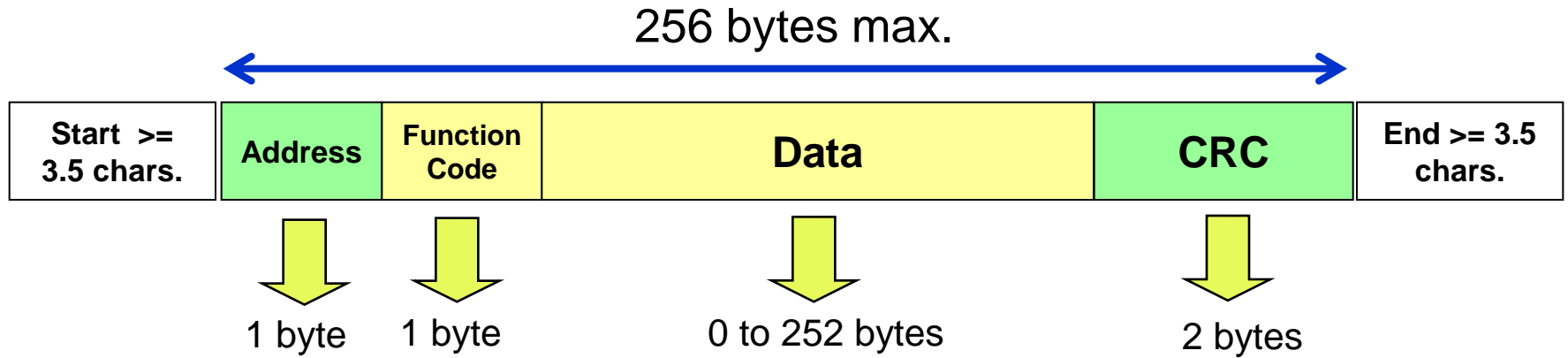


No parity check

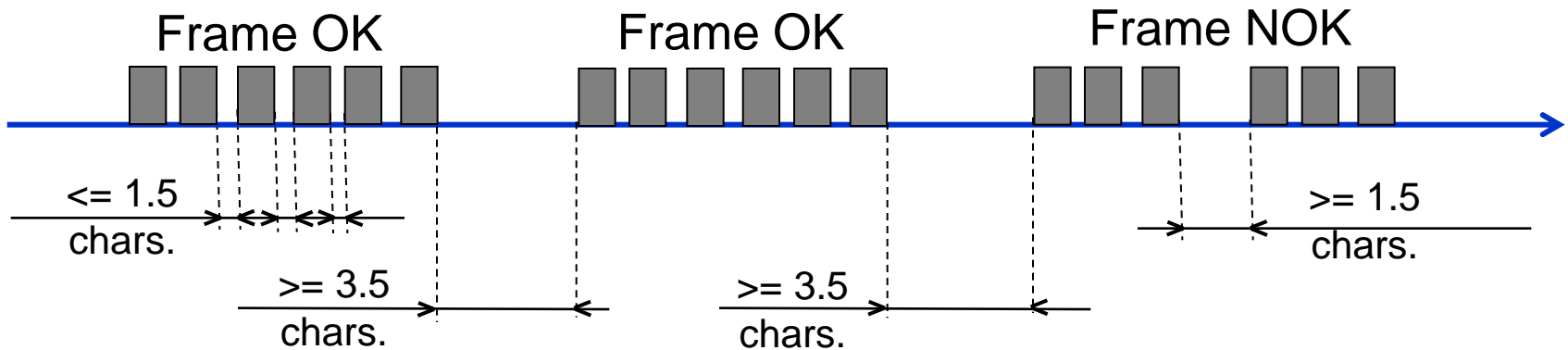
Encoding example: **Read Device Identification = 43 = 0x2B**



- RTU mode: frame format

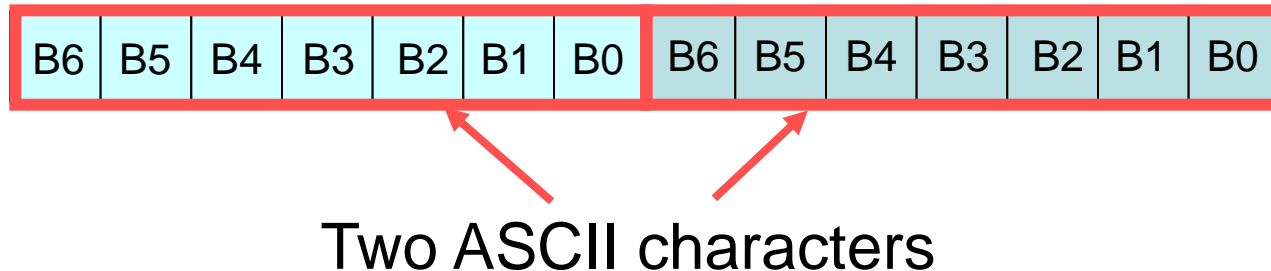


Time constraints

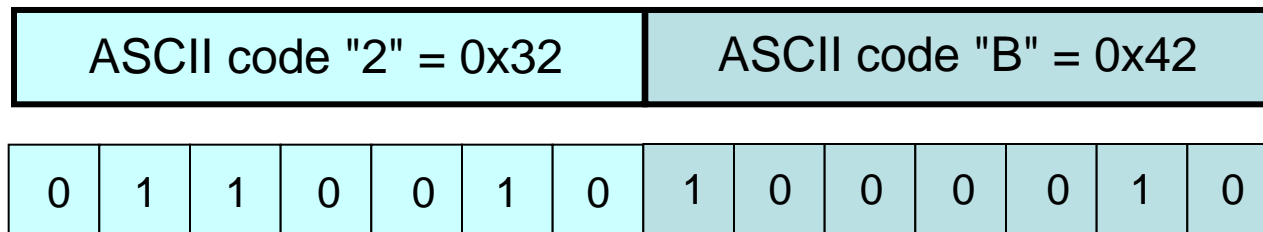


- ASCII mode: content

Every data item is encoded in **two ASCII characters (7 bits)**



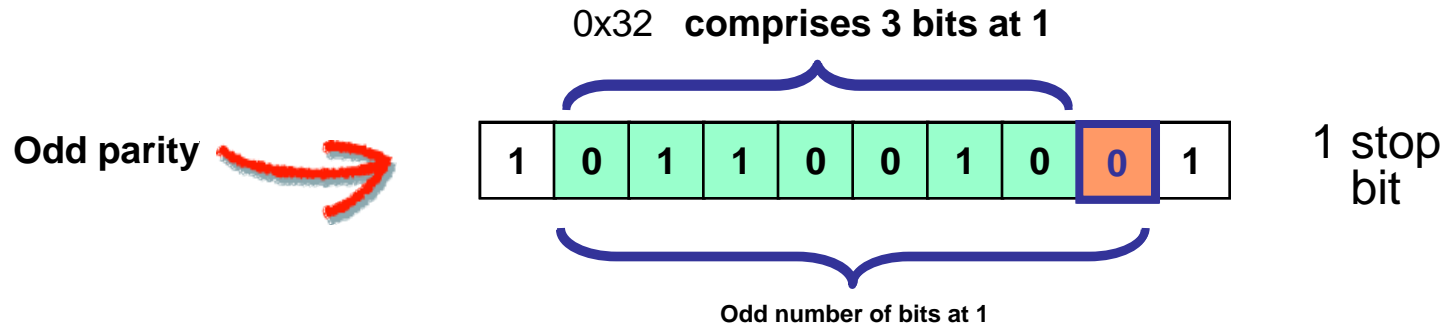
Encoding example: **Read Device Identification = 43 = 0x2B**



- ASCII mode: character format = 10 bits

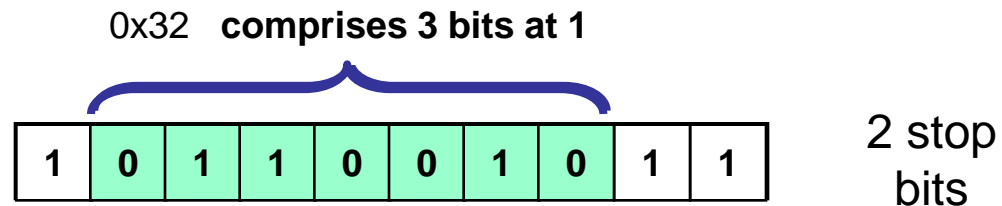
With parity check

Example: encoding character "2" in ASCII = 0x32



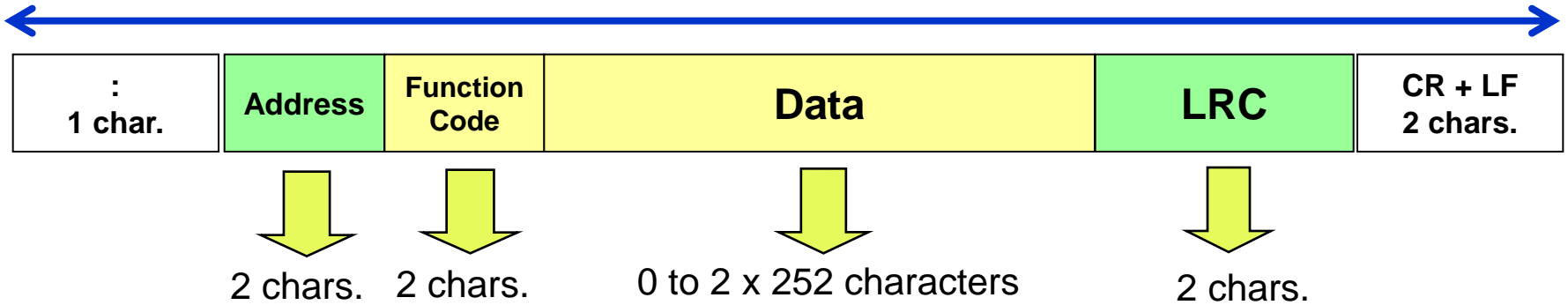
No parity check

Example: encoding character "2" in ASCII = 0x32

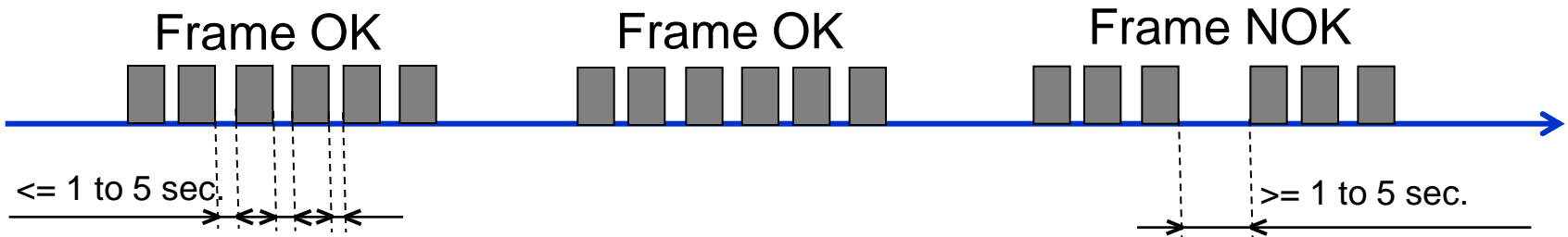


- ASCII mode: frame format

513 characters max.



Time constraints



No inter-frame time constraint.

- Transmission error checks

Two types of checks performed at the master and slave ends



Parity check

Character level

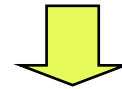


Choice of even or odd parity



Frame check

Message level



If RTU mode: calculate CRC
CRC = Cyclical Redundancy Checking

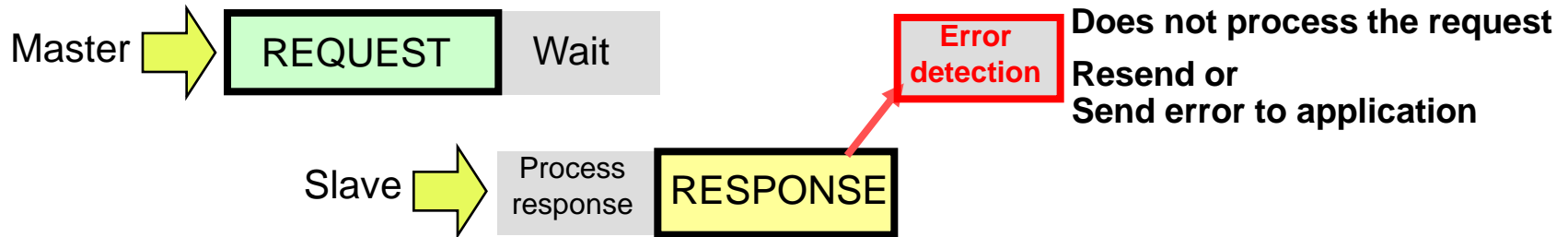
If ASCII mode: calculate LRC
LRC = Longitudinal Redundancy Checking

- Reaction on error detection

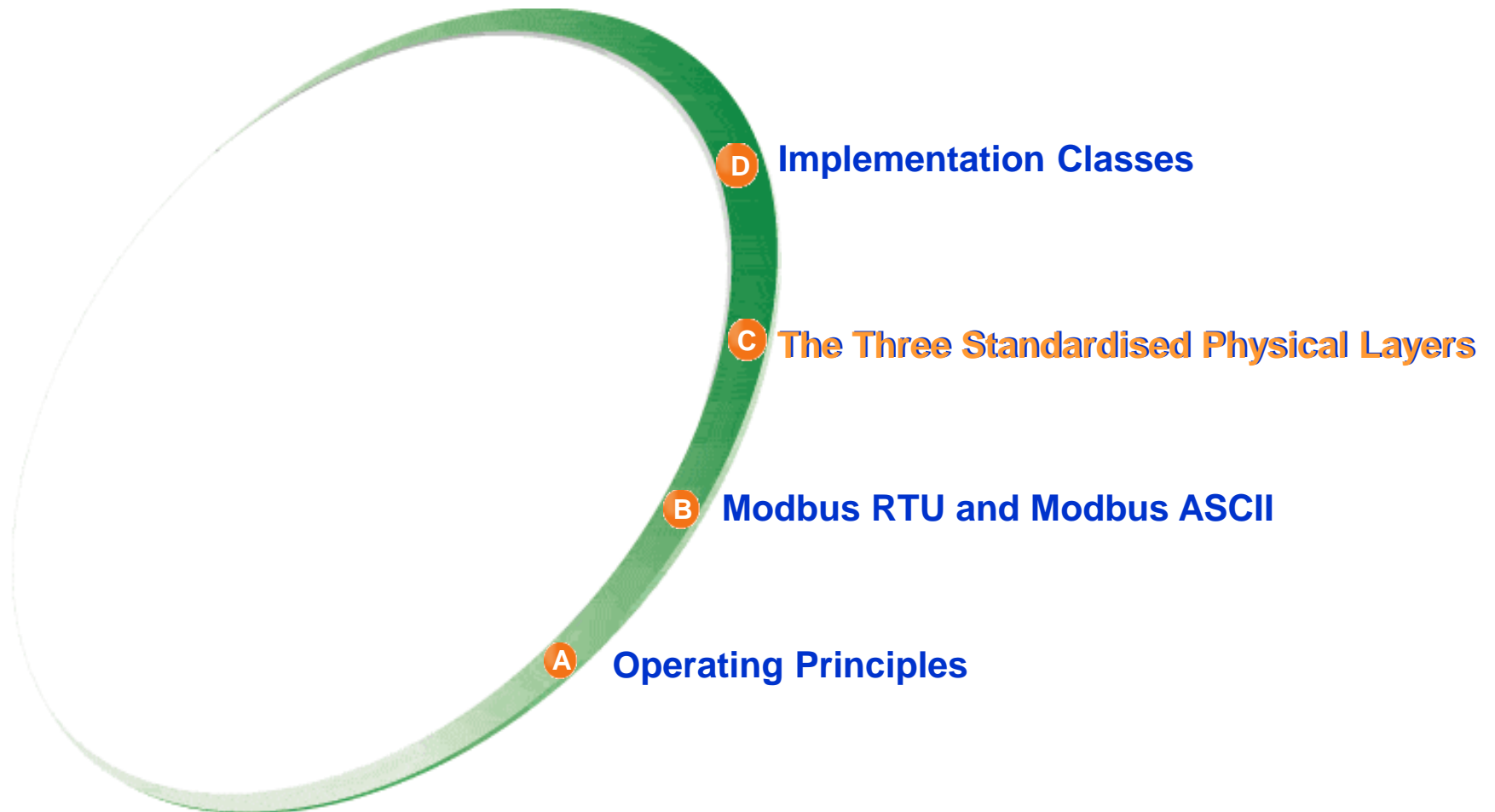
→ The slave detects a transmission error



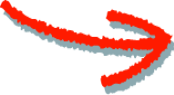

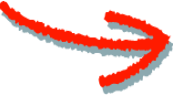
→ The master detects a transmission error



THE THREE STANDARDISED PHYSICAL LAYERS

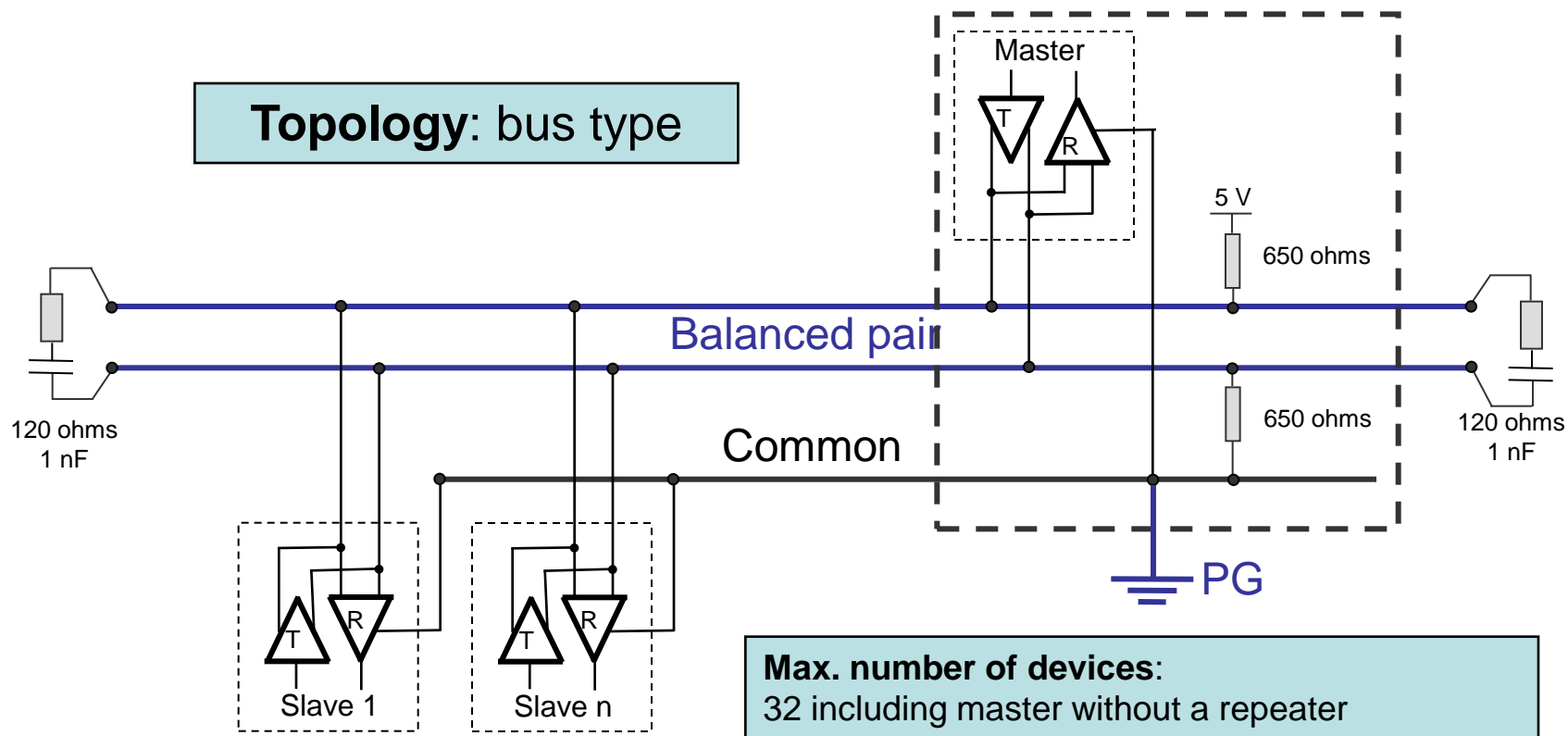


- Three standardized Modbus electrical interfaces

	RS485 2 wire	Required	Multi-point and point-to-point
	RS485 4 wire	Optional	To integrate into an existing installation without modification
	RS232	Optional	Point-to-point mode 15 meter max.

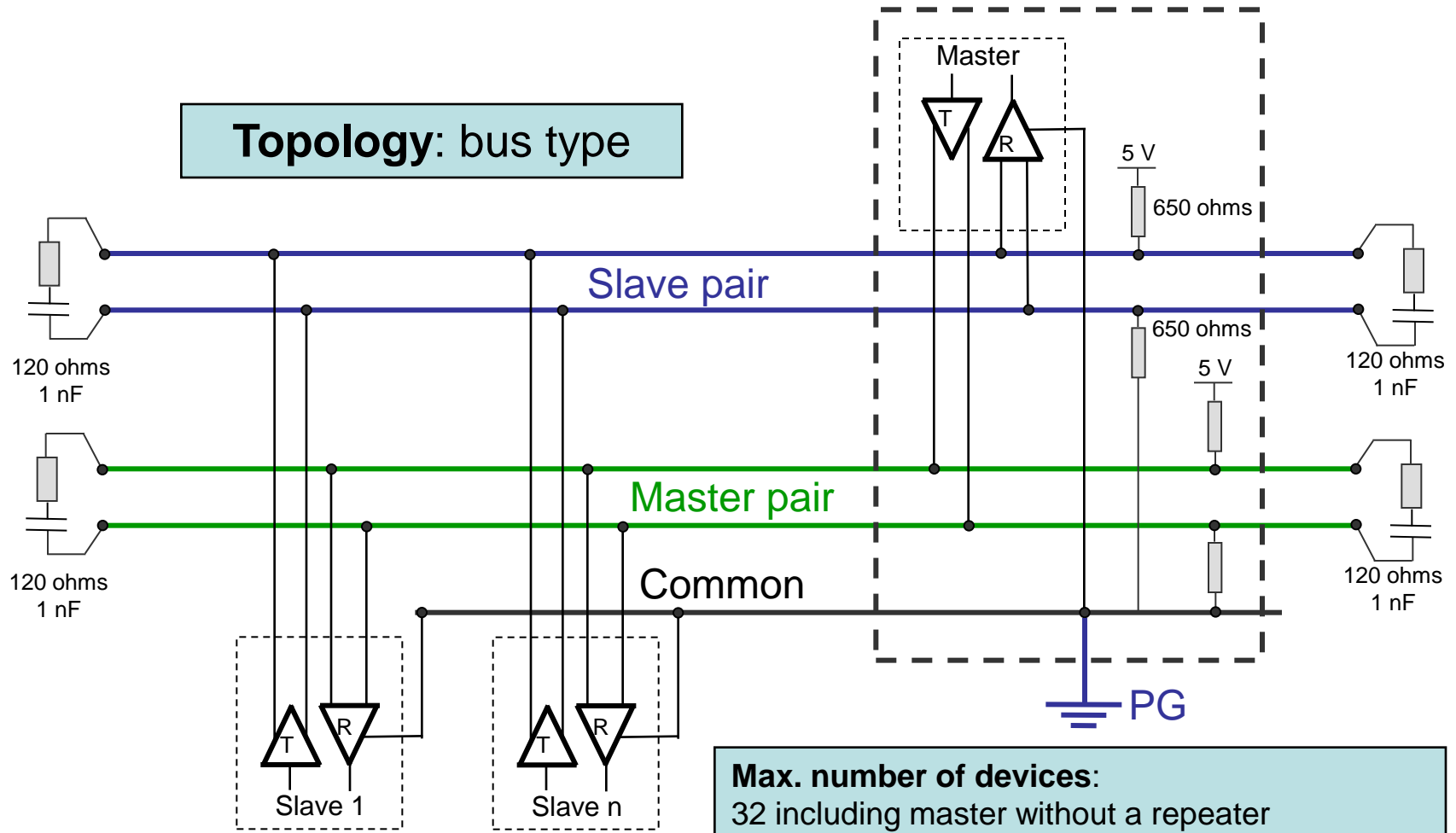
Transmission speeds:
9600 bps and 19200 bps required with 19.2 Kbps by default
Other speeds: 1200, 2400, 4800, 38400, 56 Kbps, 115 Kbps optional

- Modbus RS485 2 wire installation



Max. length:
Main segment: 1,000 meters at 19,200 bits/s
Drops: 40 meters in total

- Modbus RS485 4 wire installation

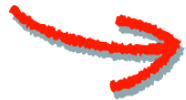


Max. number of devices:
32 including master without a repeater

Max. length:
Main segment: 1,000 meters at 19,200 bits/s
Drops: 40 meters in total

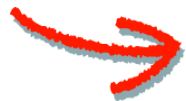
- Modbus 2 and 4 wire compatibility

Two possibilities:



Connecting 2 wire devices to a 4 wire installation

Possible, by **adapting installation wiring**

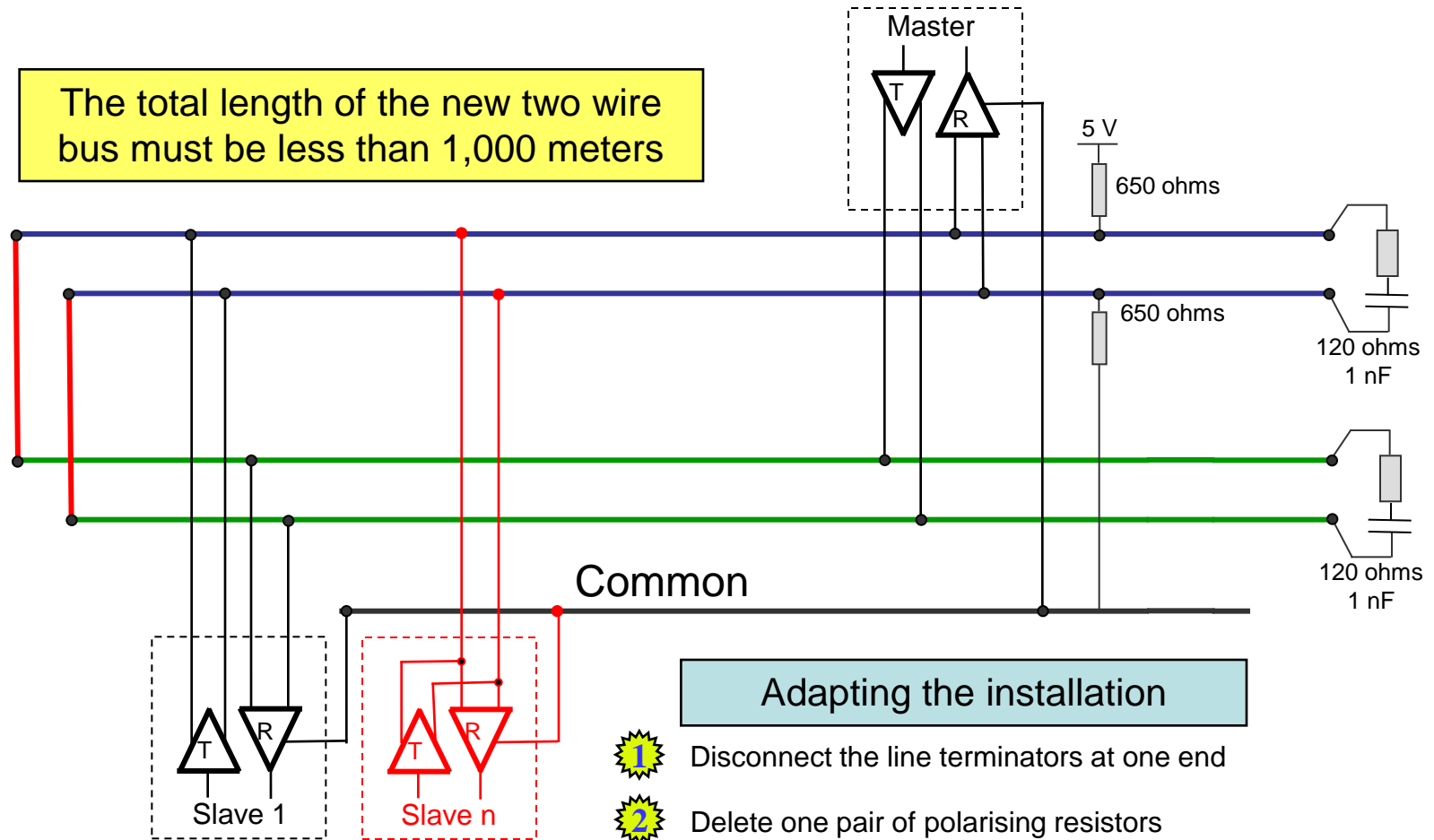


Connecting 4 wire equipment to a 2 wire installation

Possible, by **adapting device wiring**

- Connecting 2 wire devices to a 4 wire installation

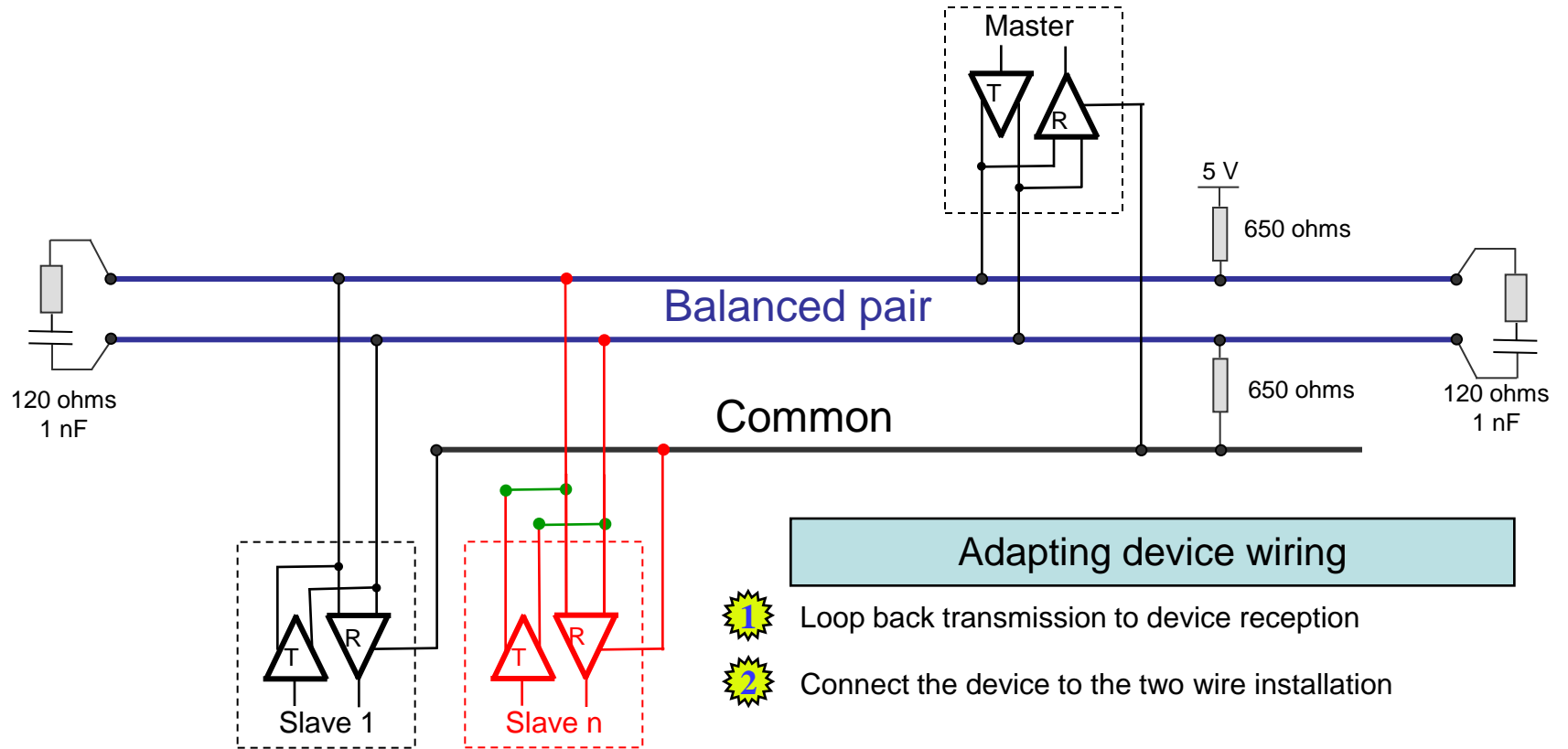
The total length of the new two wire bus must be less than 1,000 meters



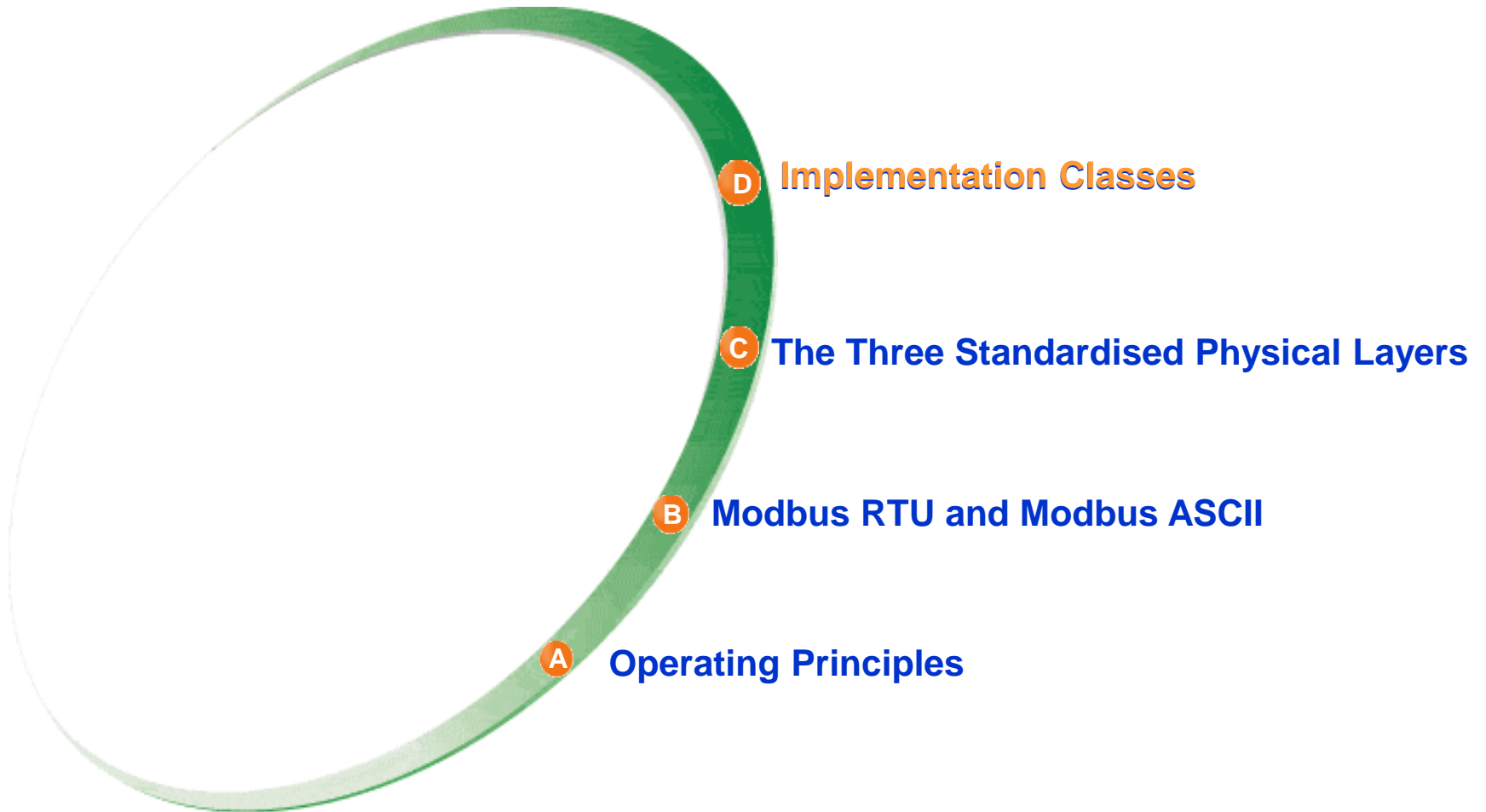
Adapting the installation

- 1 Disconnect the line terminators at one end
- 2 Delete one pair of polarising resistors
- 3 Loop back the two pairs at one end
- 4 Connect the device using two wires

- Connecting 4 wire devices to a 2 wire installation

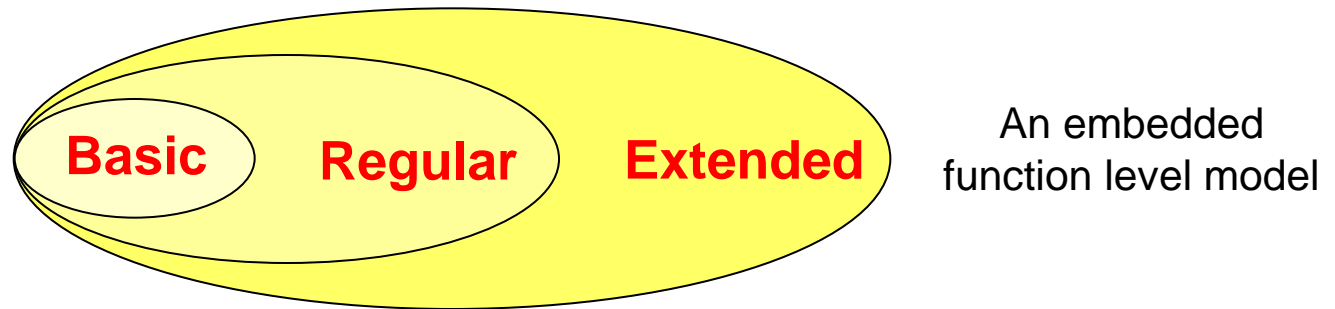


IMPLEMENTATION CLASSES



- Transparent Ready rules and vocabulary reminders

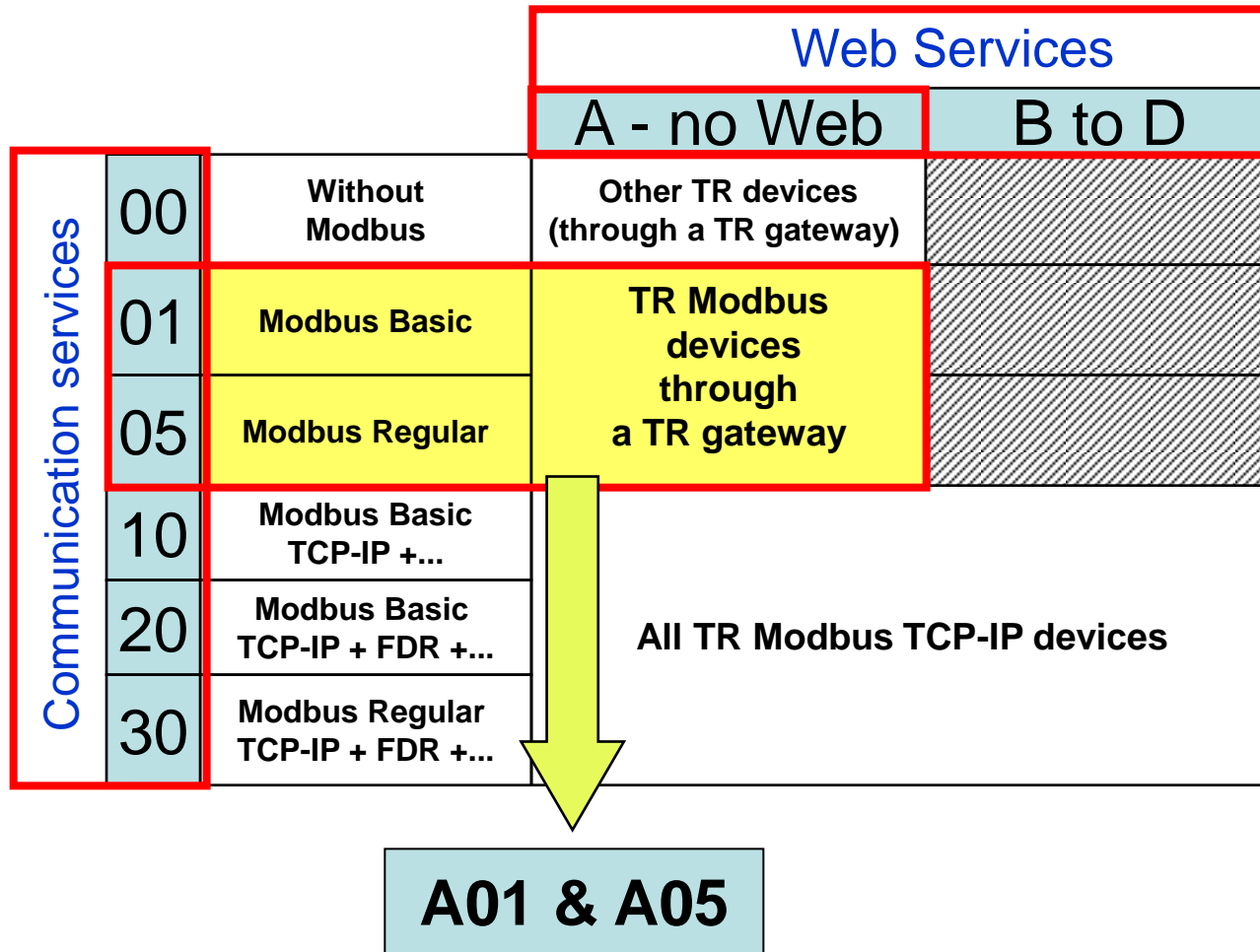
Three implementation levels for three service levels



Belonging to a class **only if** all of the required services are supported

A device **can also** support services belonging to a higher class.

- Transparent Ready equipment identification



Two implementation classes for Modbus serial line devices

- The two implementation classes

→ Class A01

Messaging class: Basic

Access to registers only

Data link class: Basic

Not configurable

→ Class A05

Messaging class: Regular

Basic + access to bits and diagnostics

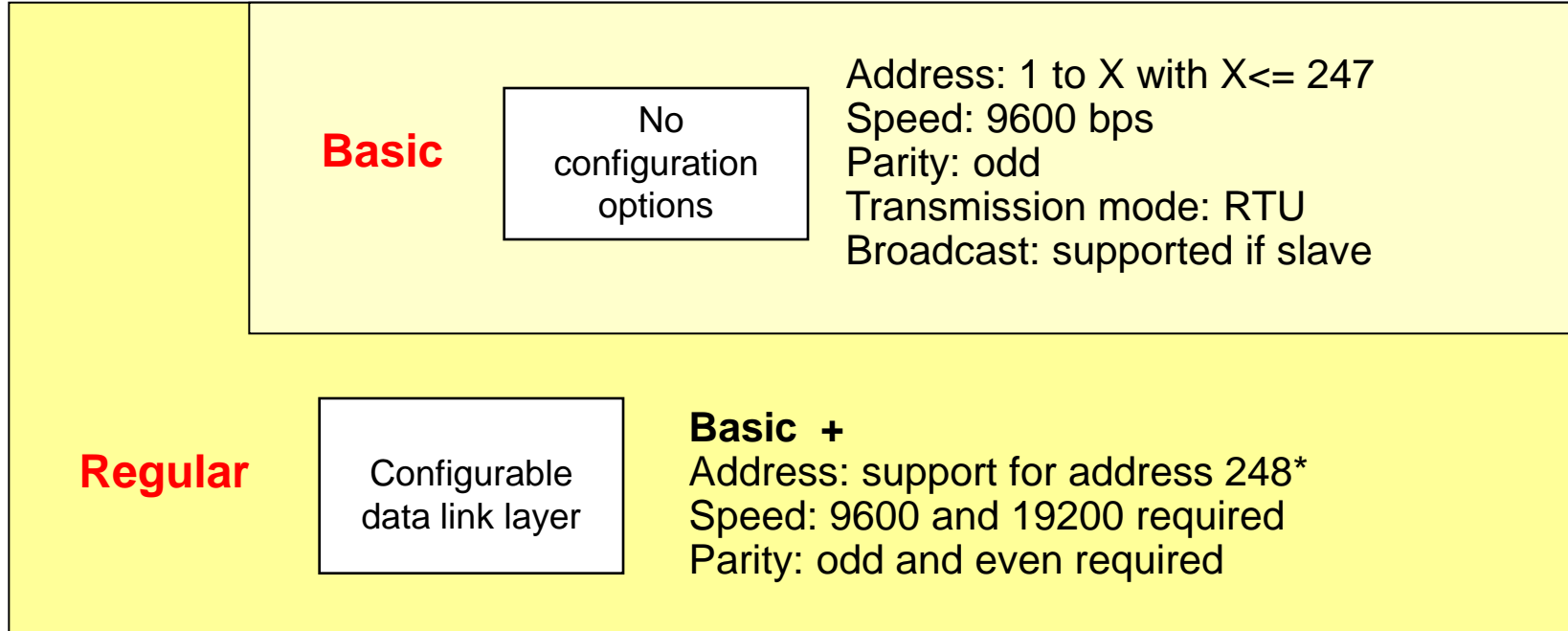
Data link class: Regular

Configurable

Device management class: Basic

Level 1 identification

- Service classes – data link layer



* Address 248 is used if the configuration can be set by messaging